PREDICTION OF INNER WOOD DEFECTS FROM OUTER BARK SHAPE

A Dissertation
Presented to
The Academic Faculty

By

Mohamed Mejri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2020

PREDICTION OF INNER WOOD DEFECTS FROM OUTER BARK SHAPE

Approved by:

Dr. Cedric Pradalier, Advisor
School of Computer Science
Georgia Institute of Technology

Dr. Ghassan AlRegib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David V Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John R Barry
School of Computer Science
Georgia Institute of Technology

Date Approved: April 18, 2020

In the strict formulation of the law of causality-"if we know the present, we can calculate the future"-it is not the conclusion that is wrong but the premise.

Werner Heisenberg

To my parents, to my brother and sisters , to my professors Thank you

ACKNOWLEDGEMENTS

I would like to thank Dr. Pradalier, the team at the DREAM Lab , and the administration at Georgia Tech Lorraine, and the Unité Mixte Internationale . This work would have not been accomplished without your support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The analysis of the internal structure of trees is highly important for both forest experts, biological scientists, and the wood industry. Traditionally, CT-scanners are considered as the most efficient way to get an accurate inner representation of the tree. However, this method requires an important investment and reduces the cost-effectiveness of this operation. Our goal is to design neural-network-based methods to predict the internal density of the tree from its external bark shape. We will compares different image-to-image (2D), volume-to-volume (3D) and Convolutional Long Short Term Memory based neural network architectures in the context of the prediction of the defect distribution inside trees from their external bark shape. Those models are trained on a synthetic dataset of 1800 CT-scanned look-like volumetric structures of the internal density of the trees and their corresponding external surface. Those different methods and approaches might potentially help in predicting the internal defect distribution of a real CT-scanned log from its external shape. However, identifying and extracting the most relevant and predictable internal defects is necessary before applying any deep-learning based method in order to predict the internal structure of the tree.

CHAPTER 1
INTRODUCTION AND BACKGROUND

For the wood industry, several studies [1][2][3] have shown that the transformation of the wood based on the internal density of the log improved the value recovery. Knots are considered as the main important inner element of the wood. They are either characterized manually or through X-Ray CT-scanning.

As an example, using X-Ray information, in a study based on several hundreds of logs of Scots pine (Pinus sylvestris L.) and Norway spruce (Picea abies L.) [2], assessed an average increase of the recovery value by 13% when compared to a sawing position of the log based on the outer shape. However, the analysis of the inner structure of the log requires highly experienced forests experts and an expensive investment (≈$5M). Furthermore, the tree bark shape reveals information about the internal structure of the tree and particularly its branching. From a living branch until a knot in duramen, these different stages have a more or less evident impact on the external bark roughness.

The arguments stated above prove the existence of a potential correlation between the internal density of the log and its outer shape. To evaluate this correlation, we propose to compare different 2D and 3D based neural network architectures. However, finding complex defects in the inner structure of a log from its internal shape using deep learning methods requires a dataset with thousands of real CT-scanned logs with their corresponding tree bark shape. Hence, a synthetic CT-scanned look-like dataset is generated. To predict the internal density of the tree, three main varieties of neural network architectures were tested, which are 2D Encoder-Decoder architectures, 3D Encoder-Decoder models, and Convolutional-LSTM (CLSTM) based Encoder-Decoder architectures. In the following chapters, we will detail and assess the different approaches we followed to predict the internal structure of the tree from its external shape using a synthetic dataset. We will take advantage of the

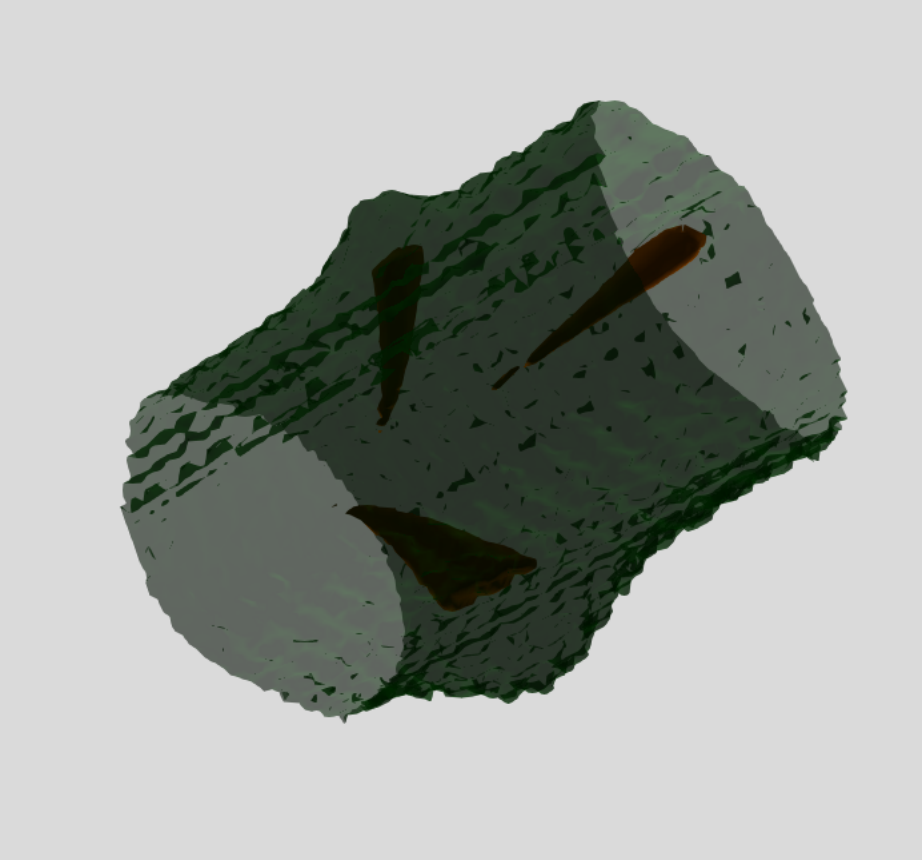

Figure 1.1: (Green) surface (Brown) iso-surface of inner density

methods and approaches we already tested on a synthetic dataset to predict the internal defect distribution of the real CT-scanned dataset from their external tree bark shape.

CHAPTER 2

RELATED WORK

2.1 Data Modeling

According to forest experts, the distribution of the internal defects in a log, especially knots, gives valuable information about the quality grade of Roundwood. Predicting such distribution from the external tree bark shape is challenging, especially when it comes to manipulating real CT-scanned cross-section of logs. Hence, building a proof of concept on a synthetic dataset is necessary. [3] tried to reconstruct a 3D volumetric simulated dataset of the log with their corresponding knots based on a real CT-scanned dataset of logs. Furthermore, [3] found an efficient method to improve the work time and yield of volumetric use and utilities of the sawmills that process the pruned logs of Pinus radiata, linking the external information provided by an industrial scanner and the simulation of defective cylindrical core (DCC) in the constitution of a three-dimensional log. The figure 2.1 is a 3D volumetric representation of a Pinus radiata log, its diameter of DCC is 10 cm, the length of its internode averaging is 70 cm. It consists of four knots centered pith simulated as truncated cones and oriented at 24° to the pith

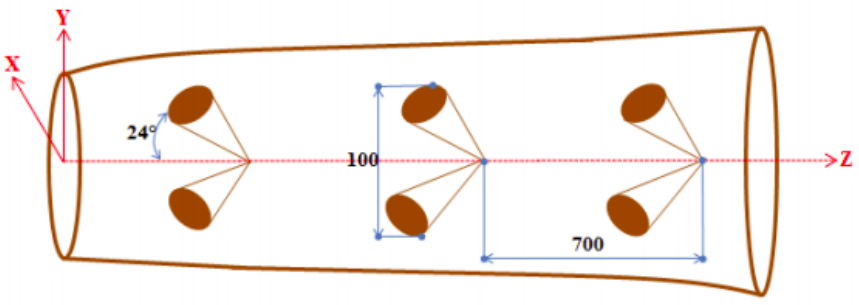

Figure 2.1: Parameters of log[3]

2.2 Internal Structure prediction

In pixel-wise regression, the typical structure of models is based on an encoder followed by a decoder composed mainly of convolutional layers. In this paper, we will focus on 2-D, 3-D, and CLSTM based encoder-decoders.

2.3 2D-models

Several 2D neural network models that aims to perform pixel-wise classification exists. [4] introduced SegNet Encoder-Decoder architecture. The encoder, a sequence of the Convolutional and Pooling layer, compresses the image while extracting the most relevant features in the data. The Decoder is a sequence of upsampling and convolutional layers that builds a pixel-wise classification map for semantic segmentation. This architecture could be useful for our project. We predicted a slice of the density within the radial longitudinal plane, using auto-encoder like architectures (e.g., SegNet [4] model). However, this problem involves regression and not pixel-wise classification. That's why the network has to be adapted, and the loss function changed to a mean-squared error rather than a sparse categorical cross-entropy. Other Neural network architectures inspired from SegNet [4] exist: U-net is a more complex model introduced by [5] that aims to feature skip connections.Other more complex architecture exists such as Fully Convolutional Networks, DeepLab [6], FCN [7], and PSPNet [8].

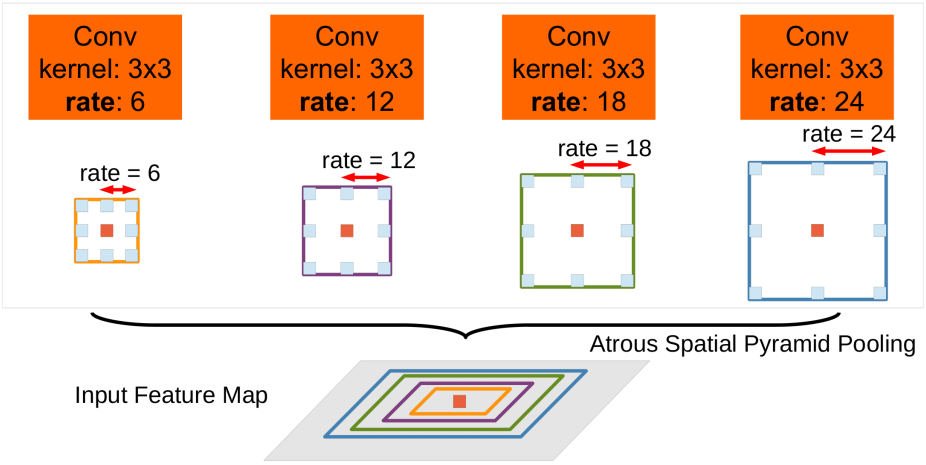

Figure 2.2: Atrous Spatial Pyramidal Pooling layer[8]

Figure 2.2 explains the mechanism of the Atrous Spatial Pyramidal Pooling (ASPP) layer. It consists of a set of 3 by 3 dilated convolution layers with four different rates (6, 12, 18 and 24) that aim to down-sample the input feature map at different scales. The output of those layers is then concatenated to improve the accuracy of the pixel-wise classification. The ASPP layer is a fundamental part of the Deeplabv3+ model. The structure of its architecture is shown in the Figure 2.3.

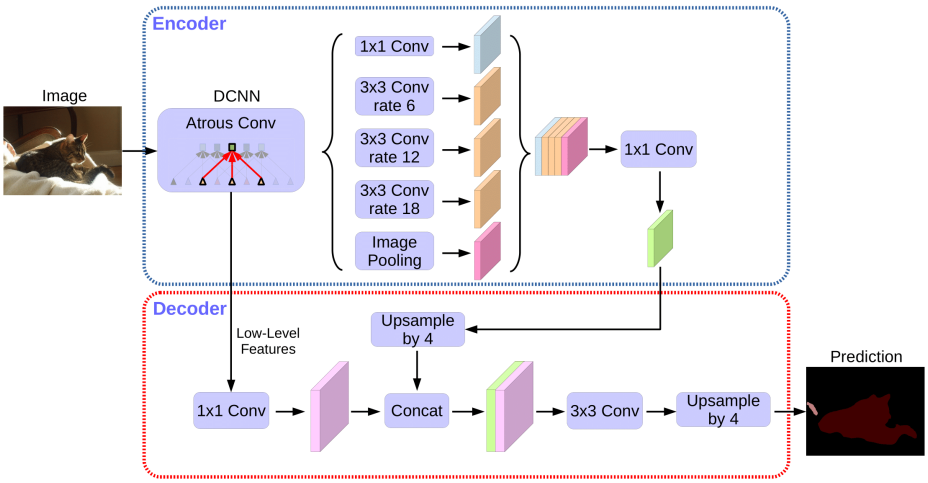

Figure 2.3: DeepLabV3+ Structure [6]

2.4 3D Models

Since the development of 3D data acquisition techniques, especially in medical imaging, several 3D neural network architectures have been designed to perform volumetric segmentation. 3D-U-Net [9] was designed to perform kidney [9] segmentation. It extends the U-Net architecture [5], by replacing all 2D operations with their 3D counterparts. The figure below summarizes the contraction and expansion paths of the 3D-UNet. However, due to computational constraints, [9] reduces the number of layers in both the encoder and the Decoder by doubling the number of channels already before max pooling and including batch-normalization. Hence, [9] ends up with a faster convergence neural network.

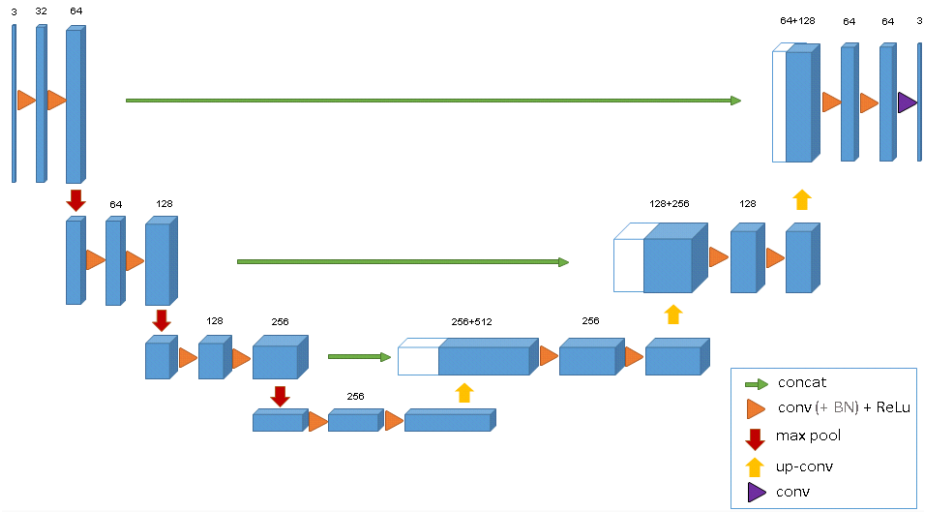

Figure 2.4: 3D-UNet Structure [9]

H.Chen [10] introduced a voxel-wise residual neural network based on residual neural network [11]. It consists of three stacked residual modules followed by four 3D-deconvolutional layers. Unlike 3D-U-Net model, VoxResNet includes skip connection layers which allows to build deeper encoder while preventing from detail loss due to the high number of convolutional blocks. This structure inspired from 2D ResNet [11] improves the accuracy of pixel wise classification for hard examples. The Figure 2.5 shows an overview of the VoxResNet model structure and a detailed representation of VoxRes module. Ac-

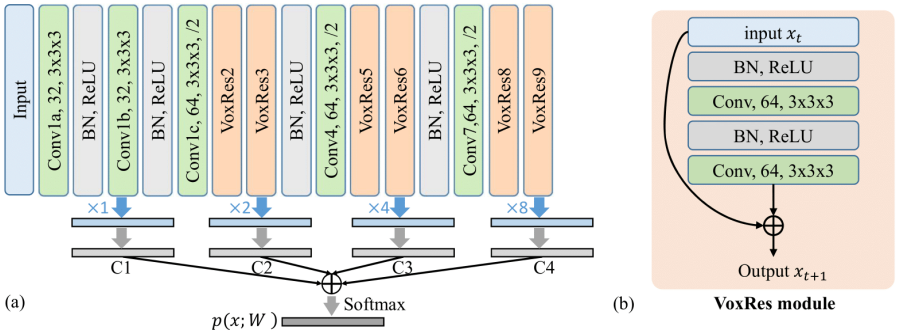

Figure 2.5: VoxResNet and VoxRes module structure[10]

cording to [10], VoxResNet achieves better results than 3D-U-Net [9] after being tested on MICCAI MRBrainS challenge data [10].

2.5 ConvLSTM-models

Both 3D and 2D encoder-decoder architectures discussed so far cannot capture the correlation that may exist between the successive elements in the dataset. Recurrent Neural Network and, more specifically, Long Short Term Memory (LSTM) structures are widely used in natural language processing to achieve sequence-to-sequence processing. Those neural networks aim to retrieve correlations between words in the same sentence. S.Nabavi *et al.*[12] introduces Convolutional-LSTMs (CLSTM) [13] based encoder-decoder structure.

The encoder structure generates feature-maps from the input images that are fed to the CLSTM [13] module. It consists of a Long Short Term Memory module where the weights are replaced with a filter bank of a convolutional layer. The Decoder is composed of several deconvolutional[14] layers and combines the outputs of different CLSTM modules and generates the segmentation map for the next time-step. Bidirectional CLSTM based segmentation structures have also been introduced and aim to capture the temporal information in both directions. the Figure 2.6 shows the architecture introduced by [12] in order to find the correlation between different frames of an images sequence while performing pixel-wise classification

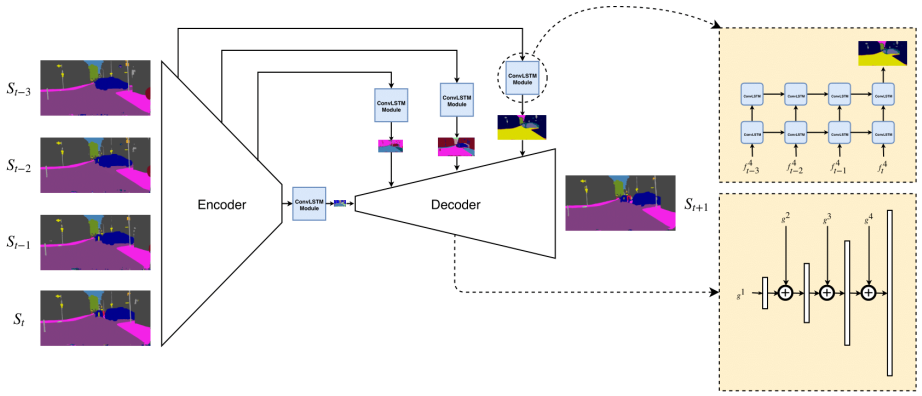

Figure 2.6: CLSTM based Encoder Decoder architecture [12]

2.6 Internal defect detection

Different object detection models exist and could be useful to detect the most relevant defects in the Roundwood (i.e., knots). These models are divided into two different categories based on their structure: two-stage object detection models and one-stage object detection models.

2.7 Two stages object detection models

The first two-stage based detector was introduced by [15]. A first neural network is trained to find several candidate boxes. Each region is then forwarded to a convolutional neural network that predicts the class score and the regression offset. [16] took advantage of this architecture and proposed a more efficient and faster object neural detection network called Faster R-CNN: it combines both anchor-based object proposal generation and region classification in the same deep neural network. The Figure 2.7 summarizes the different step followed by the Faster R-CNN algorithm

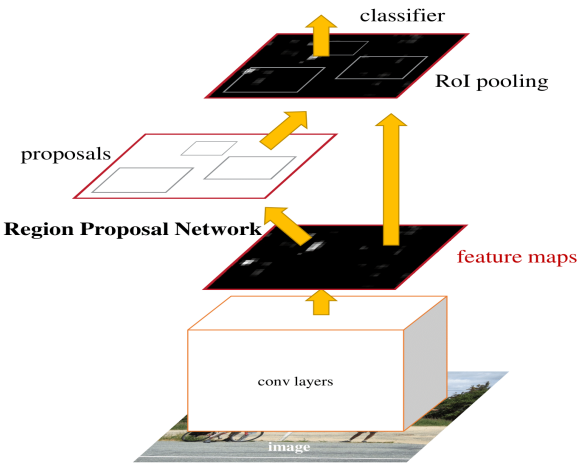

Figure 2.7: Faster R-CNN architecture [16]

2.8 One-stage object detection models

Since two stages based object detection algorithm are time and memory consuming, faster algorithms have been proposed. Single-stage based object detection models have been introduced with the Single Shot Detector (SSD) algorithm [17]. The network directly outputs a dense set of boxes and associated per class scores. The Figure 2.8 shows the architecture of the SSD model.

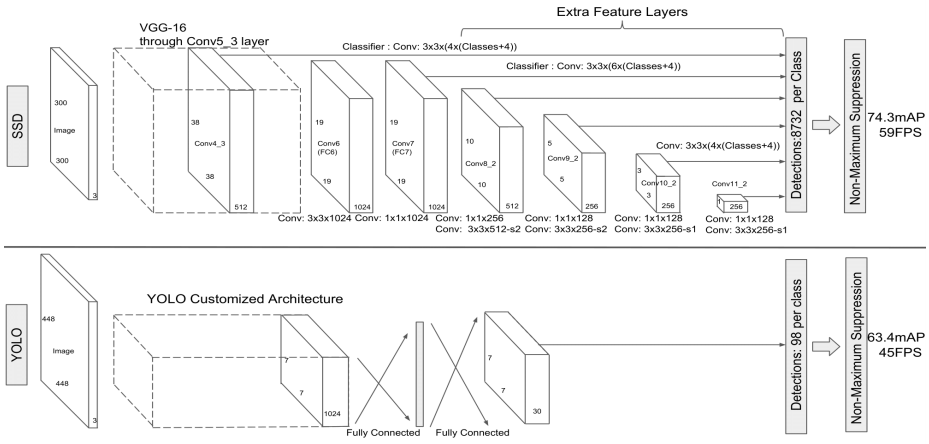

Figure 2.8: SSD architecture [17]

The main advantage of this approach is its shorter processing time which is essential when it comes to perform real time detection. However, Faster RCNN and other more sophisticated two stages models have better accuracy than the SSD [17] algorithm and other single stage object detectors .

CHAPTER 3

METHOD

3.1 Dataset

3.1.1 Synthetic Dataset

The main objective of our research is to predict the internal density of the log from its external surface. We propose to rely on deep learning architectures and modeling. However, training a complex neural network models requires thousands of logs with their internal density and external shape. Yet, the acquisition of the CT-scanned cross-sections of the tree is expensive and time-consuming. It is then crucial to create a synthetic and realistic dataset of the logs to overcome the lack of a real dataset. The dataset is composed of 1800 synthetic logs. After assessing the CT-scanned cross-section of the trees with high and low resolution, we observe that increasing the resolution from 64x64 to 512x512 is not relevant to detecting the distribution of defects.

Furthermore, training a neural network architectures on a low-resolution image is less memory and time-consuming. Hence, the log is modeled with 64 low-resolution slices (64x64 pixels) along with its height.

To get regular training, validation, and testing scores while running the neural network models, the dataset has to be balanced in terms of the number of log per branch, so it is composed of 300 k-branch logs where k varies between 2 and 7.

External surface of the tree modeling

We chose to represent the tree logs surface as a function $s(r, \theta, z)$ where θ and z refers to the polar coordinate of the tree, while r indicates the variation of the radius the cross-section of a tree log compared to a fixed radius ($r = 0.5$) cylinder. When generating the

surfaces of the trees, we have to consider the following elements:

- THE MEAN SHAPE OF THE CROSS-SECTION OF THE TREE: $s(t, \theta, z)$ can be modeled as following $s(r, \theta, z) = R(1 + Branches(r, \theta, z))$ where $R = r(1 + \mu_{texture} + \mu_{Shape_{r\theta}} + \mu_{Shape_z})$. In real world, the transverse cut of log does not have a perfect cylindrical shape but has some irregularities. We modeled them as a combination of low frequency cosine and elliptic curves.
  $\mu_{Shape_{r,\theta}} = A_{r,\theta}.cos(2\pi.f_{r,\theta}\theta + \phi) + \frac{B_{r,\theta}}{1-e_\theta.cos(\theta)}$.

- THE MEAN SHAPE OF THE TREE ALONG ITS LENGTH: Similarly, the longitudinal section of the log has some irregularities. We modeled them as a combination of a low frequency along z and an elliptic curve.
  $\mu_{Shape_z} = A_z.cos(2\pi.f_z z + \phi_z) + \frac{B_z}{1-e_z.cos(z)}$.

- THE TEXTURE OF THE TREE BARK: The texture is modelled as a high frequency cosines along θ for horizontal variations and along z for vertical variations.

- THE POSITION AND SHAPE OF THE BRANCHES: the branches which refer to the region of the defects in the trees will be modeled as Gaussian curve along z axis and θ axis where its standard deviation and mean along θ and along z depends on r. $Branches(r, \theta, z) = \sqrt{\sigma_{theta} r + \sigma_z}.exp^{-\frac{d\theta^2 + dz^2}{2}}$. Their positions and lengths are randomly set.

Figure 3.1: (Left) surface without branch (Right) surface with a branch

Internal density of the tree modelling

The internal density characterises a synthetic look-like CT-scanned cross section of the log. The objective behind generating the internal density is to model appropriately the distribution of the defects. The external shape of the tree and the internal density are correlated. Let's consider $d(r, \theta, z)$ the density of the log. The branches, a region with high density, will be modeled using square root functions on longitudinal section projection and linear functions on cross section. The number of branches in a log varies between 2 and 7 and their position and height are randomly chosen.

$d(r, \theta, z) = D.e^{-\frac{1}{2}(d\theta^2 + dz^2)}$ where $d\theta = \frac{(\theta - \mu_\theta)[2\pi]}{\sigma_\theta}$ and $dz = \frac{(z - \mu_z)}{\sigma_z}$. The figures below shows a cross section and a longitudinal section of a log.

Figure 3.2: (Right) cross section of a real CT-scanned log (ash tree) (Left) cross section of a synthetic log

3.1.2 synthetic CT-scanned look-like dataset

The prediction of the internal density of a real CT-scanned cross-section of the log is complicated for various reasons. To overcome this issue, we identified the most relevant components of a round-wood and tried to improve our synthetic dataset in a way that looks like a real CT-scanned dataset. In this section, we will try to build a look-like an external shape and internal density of an ash tree

External tree bark shape

Unlike elm or fir trees, the external shape of ash trees tends to be circular and regular (i.e., less variation in the tree bark shape along with the height of the log). The external shape of the tree is less dependant on the defect (i.e., knots) than the synthetic tree bark shape may be. Hence, we decided to reduce the amplitude of the Gaussian curve that illustrates the variation in the external surface along both the radius and the height of the tree to fit the real bark shape variation. Furthermore, we observed that the center of the round-wood depends on the height of the cross-section of the log. Hence, the center of the synthetic roundwood will depend on the height of the tree.

Internal density

The internal structure of a real CT-scanned dataset includes predictable defects (knots) as well as non-predictable defects (growth rings,...). Furthermore, the knots, the most important defects in the roundwood, have a different shape depending on the position of the cross-section of the tree and also the tree itself. According to those observations, we decided to modify the shape of the knots and add some grown rings inside the synthetic round-wood. The Figures 3.3 and 3.4 show respectively, a more realistic synthetic cross-section of a tree and a 3D volumetric representation of a modified version synthetic log.

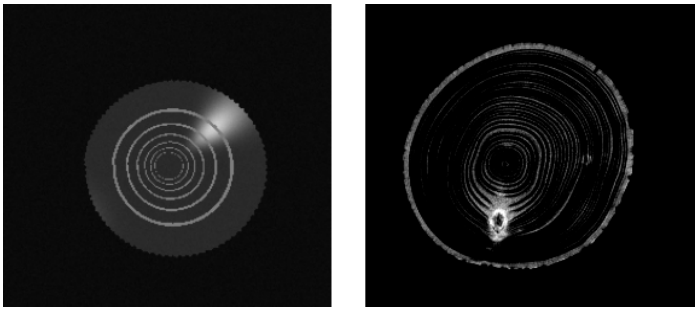

Figure 3.3: (Left) longitudinal section of a log with the growth rings (Right) cross section of a real CT-scanned log (ash tree)

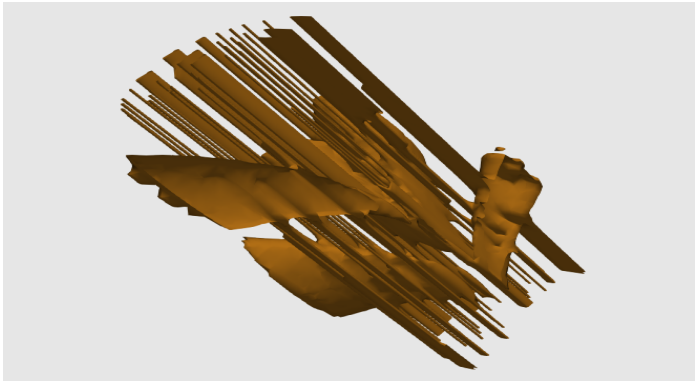

Figure 3.4: cross section of a the log with the growth rings

3.1.3 Real CT-scanned Dataset

The acquisition and the processing of real CT-scanned logs is a time consuming and difficult task for the forest experts. Hence we will manage to predict the internal density of the logs from their external shape based on a few examples.

External tree bark shape

To extract the external shape of the tree, we used a modified version of the edge detection algorithm. Common edge detection models such as the canny algorithm combine filtering to other refinement edges processes to detect the edge appropriately. However, applying that algorithm failed to extract only the tree bark shape since it considers both the tree bark and the growth rings of the log as a potential edge. Simple thresholding along different axes crossing the center of the tree extracts the tree bark shape accurately.

The figure below shows a roundwood of elm and an ash tree with their corresponding tree bark edges.

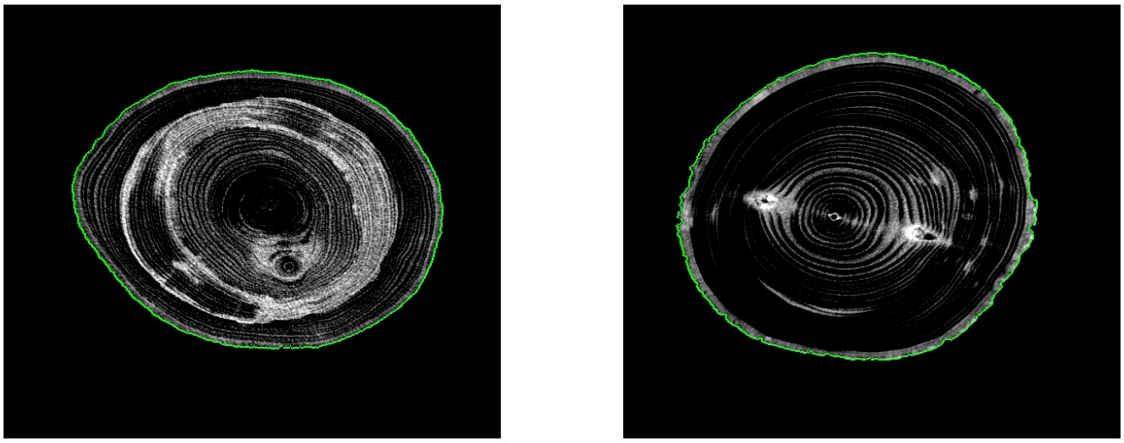

Figure 3.5: (Left) elm roundwood with its external shape (Green) (Right) ash roundwood with its external shape (Green)

considering the tree bark shape as an edge on the CT-scanned cross-section of the log fails to characterize all the relevant information about the external shape of the tree. We

considered the tree bark shape as a relative distance between a reference circle that fits the roundwood and the edges of the tree bark. The center and the radius of the reference circle can be found as follows:

$$(xc^*, yc^*, r^*) = argmin_{xc,yc,r}(\sum_{i\in\zeta} \|zi - f(zi)\|)$$

where $zi \mapsto f(zi)$ associates each point on the circle with its corresponding pixel of the tree bark edge.

Internal defect extraction

Many features on the cross-section of the log are not predictable from the external shape of the tree, and hence they are considered as a noise such as the growth rings of the tree as well as the mechanically generated defects due to cutting procedures. Hence, we try to extract the knots, the most relevant, and predictable defects inside the log. Two Stages of object detection algorithm (Faster RCNN[16]) were used to identify the knots inside the log where each defect (ie, knot) is represented by a bounding box. Since we have only one class to label and the images don't include too complicated details, the Faster R-CNN [16] model doesn't require a significant number of CT-Scanned roundwood samples for training. However, to achieve satisfactory performance despite the small amount of available hand labeled images, a dataset for each type of tree has to be generated.

3.2 Internal density prediction

This section details the different strategies and approaches used to predict the internal structure of the tree from its external shape.

3.2.1 2D-modelling

According to the figure in the first step, we have to generate 2D density layer slices from an entire log. To achieve that, each tree bark is split into 64 longitudinal stripes with a fixed

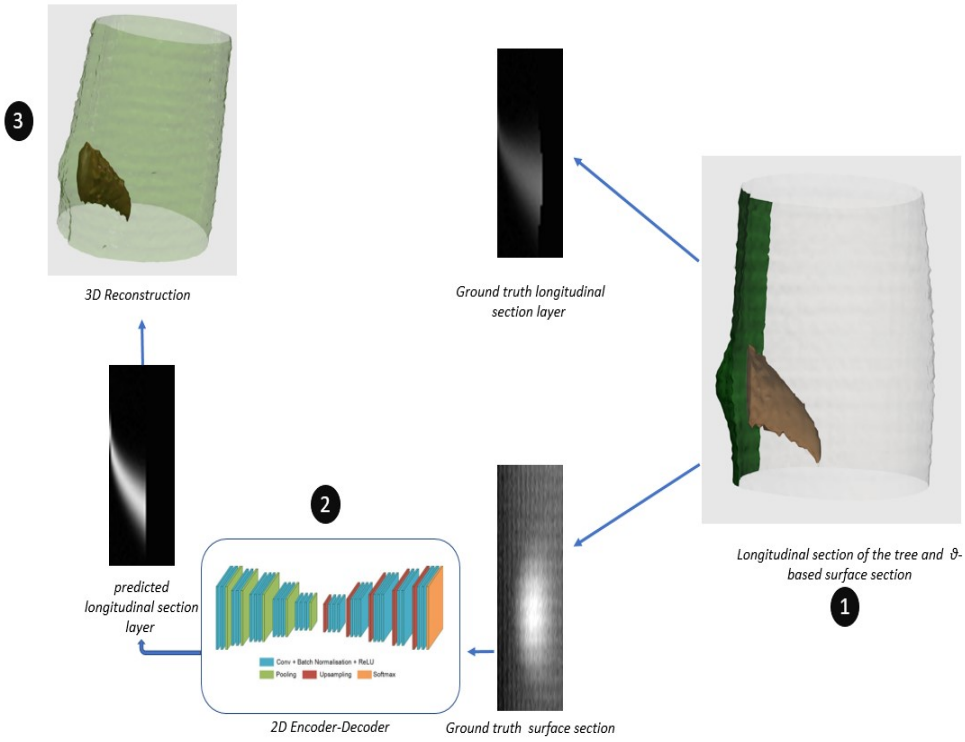

Figure 3.6: 2D Density Prediction Strategy

circular arc measure ($\theta = \frac{\pi}{6}$). Each surface strip is associated with a longitudinal layer of the internal density of the log, as shown in the figure. We end up with a 115200 size 2D dataset of (64x64) resolution density layer and their corresponding external surfaces. As stated in the previous section,three variety of 2D-Encoder decoder will be used : SegNet [4], U-Net [5] and DeepLabV3+ [6].

SegNet

The prediction of the density layer from its corresponding surface is a regression problem, so we used a mean-squared error function instead of categorical-cross entropy and removed the last softmax layer. Other loss functions like Huber loss have been tested, but the training and validation scores didn't change significantly since the output value is bounded between 0 and 1, and the main difference between mean squared error and Huber loss is for the high output value. We changed the depth of Encoder and the decoder of the original SegNet

[4] due to the low-resolution input to avoid small bottlenecks and prevent the model from overfitting. Instead of using the entire structure of a VGG-16 [18] as an encoder which is more adapted to larger images, we removed its last convolutional block. Furthermore, we replaced the rectified linear function (ReLU) with parametric-rectified linear function $P - ReLU$[19] which is an optimized-slope value version of leaky-ReLU. It improves model fitting while avoiding over-fitting issues. We use the same optimizer of the original SegNet [4], which is stochastic gradient descent with Nesterov momentum. We chose a learning rate of 10^{-3} and a momentum of 0.99, and we reduced the dropout rate from 0.5 to 0.1.

U-Net

As for SegNet[4] architecture, the loss function is changed from categorical cross-entropy to mean-squared error. For the same reasons, the contracting path and the expanding path were reduced: the 1024 filters convolution layer was removed. The activation function and the hyperparameters of the optimizer are the same as SegNet [4] architecture.

DeepLabV3+

The DeepLabV3+ [6] architectures contains much more innovative approaches and layers (Atrous Spatial Pooling Pyramids, Xception blocks, Depth-wise separable convolutions,etc) compared to U-Net [5] and SegNet [4]. No significant changes in the architecture were made. However, the hyper-parameters of the optimizer, the loss function, the regularization strategy, and the activation function were modified: The learning rate of SGD optimizer was tuned to 10^{-3} instead of 5.10^{-3}. The Mean squared error replaces the categorical cross-entropy as a loss function. We replaced the rectified linear function (ReLU) $f(x) = max(0, x)$ with parametric-rectified linear function $P - ReLU$. In addition to the L2 regularization layers already present in the original architecture we added a layer of dropout for each layer to further prevent overfitting,

3.2.2 3D-modelling

In this section, we will represent the internal density of the log as an entire voxel space. The surface is modeled as a difference of radius between the radial coordinate of a point in polar space and the radius of its projection on the external shape of the tree. According to the previous section 3 different variety of deep learning models will be tested : 3D-U-Net [9], 3D-SegNet and VoxResNet [10].

3D-SegNet

3D-SegNet is inspired by 2D-SegNet [4] by replacing the 2D layers with their 3D counterpart and removing convolutional layers in the Encoder and the decoder to reduce the memory and time expenses and prevent overfitting. Figure 3.7 shows the structure of 3D SegNet. it consists of a transformed VGG-16 [18] based Encoder with ten 3D Convolu-

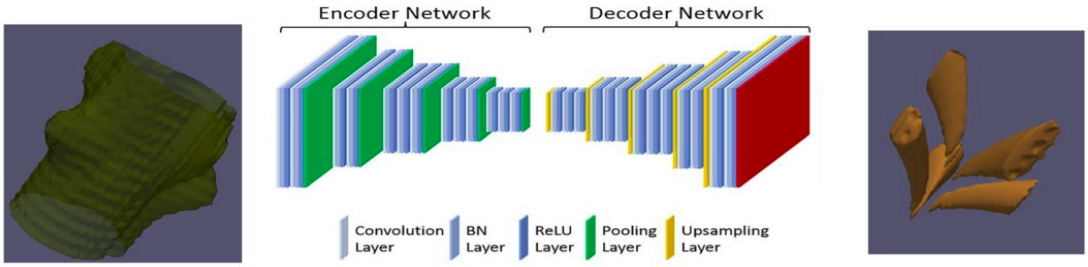

Figure 3.7: 3D Segnet Structure

tional layers. The last three 512 x Convolutional layers have been removed. Each encoder layer has a corresponding decoder layer, and hence the decoder network has ten layers. Each convolutional layer is followed by the Batch-normalization layer, and a Parametric-RelU activation layer 3D-max-pooling layer with 2x2x2 strides to avoid overlapping is used to down-sample the 3D feature map. Similarly, in the decoder 3D-Up-sampling layer performs 2x2x2 up-sampling while keeping max-pooled points at the same positions. To prevent overfitting, a dropout of 0.1 is added at the end of each convolutional layer.

3D-U-Net

We kept the original structure of 3D-U-Net [9] but made some changes in the loss function and overfitting strategy. The original 3D-U-Net [9] is trained on partially annotated voxels. Hence, the loss function has to be weighted with zero weight when the data is not annotated and one otherwise. In our case, we are performing regression with a fully-annotated dataset; we used unweighted mean-squared as a loss function. Furthermore, to prevent overfitting, a dropout of 0.1 was added after each 3D convolutional layer.

VoxResNet

One advantage of VoxResNet [10] over 3D-U-Net [9] is the ability to build deeper encoder and decoder networks while avoiding overfitting. We didn't make a significant change in the original architecture. We removed the final classification layer and kept the multi-level contextual information consisting of 4 3D-deconvolutional layers with respectively 1, 2, 4 and 8 degrees of strides. A dropout of 0.1 is added after each convolutional layer to prevent overfitting.

3.2.3 Convolutional LSTM based modelling

Conv-LSTM based SegNet

We modified the original architecture for time and memory expenses purposes. The SegNet [4] structure is used to build the Encoder and the decoder of this model. To capture the correlation between the different cross-section of the internal density of the log while reducing the computation costs, we only keep the conv-LSTM[13] layer at the end of the Encoder (bottleneck). To avoid overfitting, a dropout of 0.1 was added after each convolutional layer and inside the Conv-LSTM[13] Layer. Practically, building a convolutional-LSTM neural network is challenging due to image embedding management. To overcome this problem, we used a wrapper called TimeDistributed Layer which enables applying convolutional op-

erations multiple time to multiple input time steps and hence provides a sequence of feature maps to the LSTM model to work on.

Bidirectional Conv-LSTM based SegNet

Intuitively the correlation between a successive cross-section of log exists in the forward and backward directions. To capture the correlation in both directions, the bidirectional Conv-LSTM version of SegNet [4] could be useful. It has almost the same architecture as the Conv-LSTM based SegNet [4]. We used the same set of hyper-parameters than before. The main difference consists of a Bidirectional layer that wraps the old Conv-LSTM layer. However, to prevent overfitting, we decided to increase the dropout from 0.1 to 0.2 inside the Conv-LSTM layer.

CHAPTER 4

RESULTS AND EXPERIMENTS

4.1 EXPERIMENTS

We split the synthetic dataset into three subsets: training set, validation set, and testing set with the respective proportion 80%, 4%, and 16%. Each subset is composed of k-branch logs in the same proportion where k varies from 2 to 7. All the different neural networks cited so far were trained, validated, and tested on these subsets. For computational constraints, the batch size is fixed to two volumes, 100 images, and a sequence of 64 images for respectively 3D models, 2D models, and CLSTM based models. Each model is trained for 50 epochs. At the end of each epoch, a validation test is performed, and we save only the weights of the model with a minimum validation error. Once all models have been validated on the synthetic dataset as a proof of concept, we start predicting the internal structure of real logs from their external shape. However, the CT-scanned Roundwood dataset consists of high-resolution images (ie, 512x512 pixel per image), which might improve the defect distribution prediction. For computational purposes, we chose to train our models on a low-resolution version (ie, 64x64 pixel per image) of this dataset. The real CT-scanned dataset consists of a cross-section of four trees: ash tree, elm tree, fir tree, and aspen tree, with a different number of cross-sections per tree. As a preprocessing procedure, we removed all the images that don't contain Roundwoods. 30% of the Roundwood images of each tree are annotated manually (i.e., each bounding box refers to a knot). A pretrained Faster R-CNN, on MS COCO dataset [20], is then trained on the previously annotated dataset. This model is then used to perform coarse automatic-labeling on the 70% remaining CT-scanned cross-section images.

4.2 RESULTS

4.2.1 Architectures

In this section, we will assess the prediction of 280 synthetic logs with their corresponding external surface. To assess the training and validation error, we use a mean-squared error. For the testing process, we used the root mean squared error (RMSE). For the 2D models, there are two possible ways to make the assessment either with the predicted density layers or with the reconstructed volumetric structures. Practically, there is no significant difference between the two approaches. Table I shows the average performance of the different 2D,3D, and CLSTM based neural network architectures when tested on 40 k-branch logs ($k \in \{2, 5, 7\}$)

Table 4.1: Best results for each type of architecture

Architecture	RMSE (10^{-2})			Parameters
	2 $Branches$	5 $Branches$	7 $Branches$	
SegNet	**1.27**	**1.66**	2.46	34 M
U-Net	1.33	1.68	**2.42**	36 M
DeepLabV3+	3.17	3.23	3.48	42 M
3D-SegNet	2.39	3.13	3.92	144 M
3D-U-Net	2.48	3.10	3.70	114 M
VoxResNet	2.86	3.37	3.92	35 M
CLSTM-SegNet	2.93	4.38	5.03	31 M
Bidir-CLSTM-SegNet	2.49	3.4	4.23	52 M

We conclude from Table I that SegNet [4] achieves better prediction results than DeepLabV3+ and U-Net for two-branches logs. In contrast, U-Net [5] achieves better results for predicting the internal density of logs with a high number of branches. Those results are expected because, unlike the SegNet [4] model, the U-Net [5] architecture focuses more on capturing the information lost through the encoder structure, which may be relevant when it comes

to more complex log structure (i.e., logs with a high number of branches). DeepLabV3+ [6] fails compared to the other architectures to achieve good results: The DeepLabV3+ [6] architecture is significantly more complex compared to SegNet [4] and U-Net [5] (42M parameters for DeepLabV3+[6]) which may lead to an overfitting. Furthermore, the up-sampling and downsampling structures of the DeepLabV3+[6] fail to guarantee a better density layer reconstruction.

3D-SegNet and 3D-U-Net [9] achieve better performance than VoxResNet [10]. The 3D-SegNet model keeps the same indexes of maximum voxels value after upsampling. Unlike 3D-U-Net [9] and VoxResNet [10], 3D-SegNet doesn't contain real skip-connection or concatenation layer and hence doesn't recover information lost at the end of the encoder. This can be crucial when it comes to a more complicated structure, i.e., log with a high number of branches.

To capture the correlation between successive cross-sections of the same log, we tested two versions of the CLSTM based SegNet. Intuitively, building a model that considers a bidirectional correlation between consecutive cross-sections of a log is more relevant than observing the correlation between the different slices of the log in one single direction. This intuition is confirmed according to the RMSE results shown above. According to table 4.1, 2D-models achieves the least RMSE scores while reducing the memory expenses.

4.2.2 3D visual results

We used $Paraview^{©}$, an open-source multi-platform application for 3D visualization, to assess the 3D rendering of the predicted iso-surface of the internal density of the tree. The figures below show the predicted iso-surface with different models for the 6-branch log. Based on the RMSE evaluation, we chose to assess only the 2D, 3D, and CLSTM based models with the highest performance. We conclude that the 2-D models achieve better results than 3-D models in terms of iso-surface correspondence between the ground truth (Green volume) and the predicted log (Red volume).

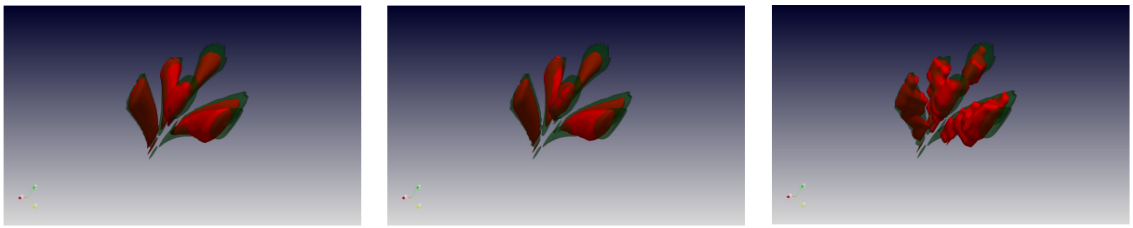

Figure 4.1: (Left) 2D-SegNet ,(Middle) 2D-Unet ,(Right) DeepLabV3+

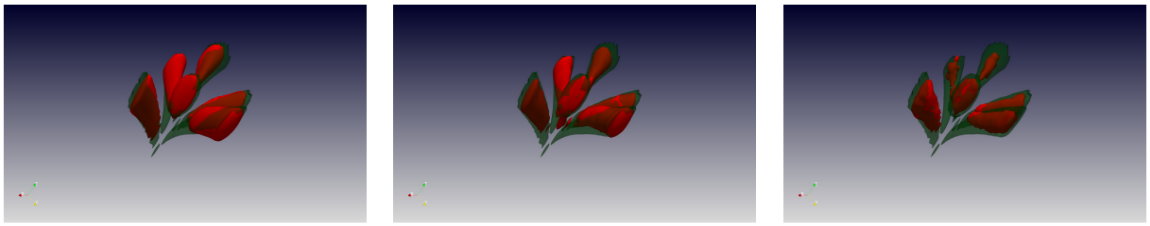

Figure 4.2: (Left) 3D-SegNet ,(Middle) 3D-Unet ,(Right) VoxResNet

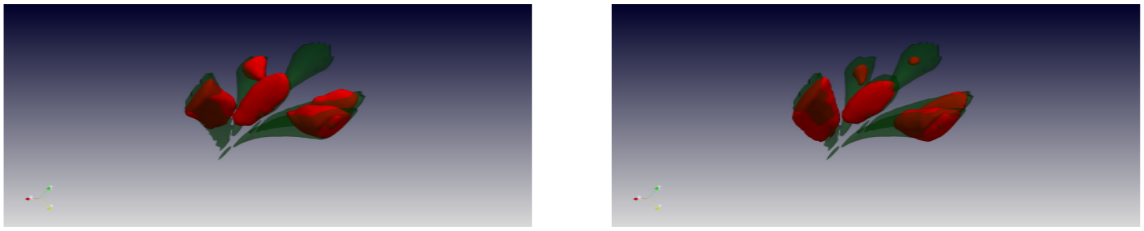

Figure 4.3: (Left) Conv-LSTM based SegNet (Right) Bidirectional Conv-LSTM based SegNet

4.2.3 Real CT-scanned density prediction assessment

As stated in the previous section data preprocessing is required to extract the most relevant (i.e. predictable) defects. A faster R-CNN model is used to identify the knots, the predictable defects inside the tree. It achieves compelling validation results as can be seen to the Table 4.2. The results shown in the Table 4.2 are confirmed after visual assessment on

Table 4.2: Detection results on four different trees

metrics	elm tree	ash tree	fir tree	aspen tree
mAP (%)	84.6	75.8	91.5	81.3
mean classifications score (%)	76.8	69.4	90.2	71.9

some unlabelled roundwood of four different trees. Hence, we used this model to perform

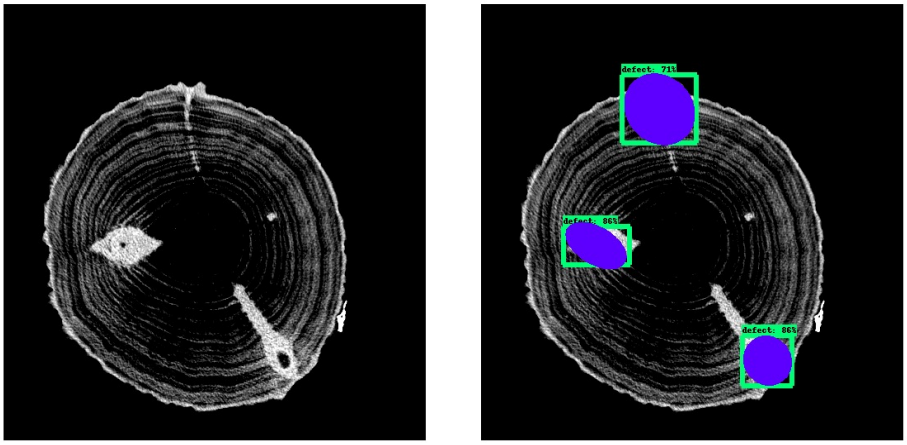

Figure 4.4: (Left) original CT-scanned cross section of an elm tree (Right) detected defect inside the CT-scanned cross section of an elm tree

coarse automatic labelling on all the test set. The figure below shows a roundwood of an elm tree with their corresponding defects (knots) identified by ellipses inside the bounding box predicted by the Faster R-CNN model.

The extracted knots represented by ellipses are used to reconstruct the tree. The figure below shows a volumetric representation of an elm tree composed of a sequence 880 cross section image. Each roundwood is a low resolution image (i.e. 64x64).

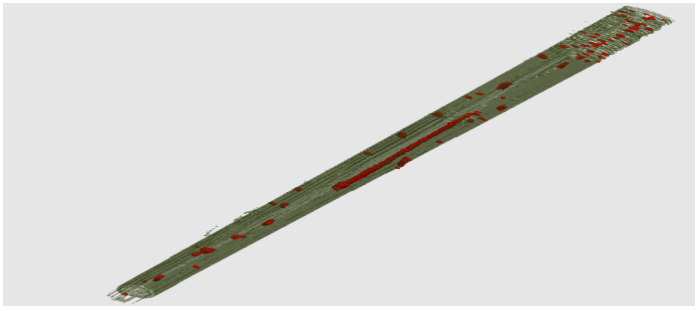

Figure 4.5: 3D representation of the external surface (Brown) and its corresponding internal density (Red) of an elm tree

Due to the scarcity of real CT-scanned logs of each kind of tree (elm, ash, fir, aspen), we can only visually assess the performance of the models on a small part of the tree. Currently, we observe that, unlike predicting the internal density of synthetic logs, which are easy

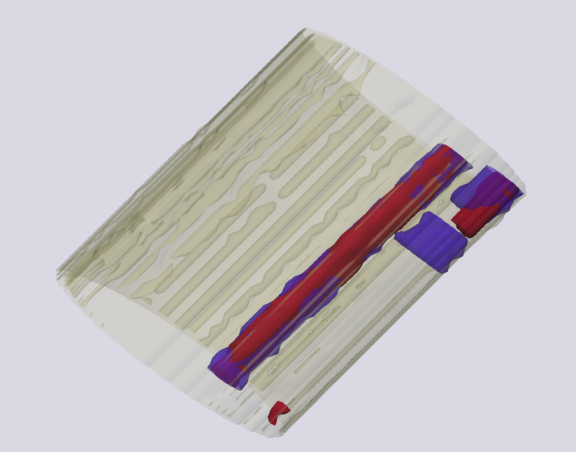

Figure 4.6: (Purple) internal density ground truth iso-surface (Red) predicted internal density iso-surface (Brown) external surface of the tree

to learn, the real data are more complicated. This could explain why only a Bidirectional CLSTM [13] based models achieve a satisfactory result.The figure below shows the ground truth and the predicted 3D volumetric sequence of 32 cross-sections of an elm tree using a Bidirectional CLSTM [13] Encoder decoder architecture, after performing the extraction of the predictable defects (knots) inside it.

Unlike the synthetic dataset, where the internal density of the logs are highly correlated with the external surface and where a simple 2D encoder decorder architecture linking the bark shape of the tree with its internal density acheives satisfactory results as stated in the previous sections, the relation between the internal defect on the real CT-scanned dataset and its external shape is much more difficult to define. Hence, a common 2D or 3D model isn't able to catch that correlation. However, few roundwood internal structures might be predicted from their corresponding external surface, a spatial correlation between defects distribution along a sequence of roundwood still exists and has to be wisely exploited. Furthermore, the predictability of the internal density of the tree from its external surface may vary with the tree itself. The figures below show the different randomly chosen roundwood of 4 different trees. However, according to the Figure 4.7, the internal defects inside the fir and aspen tree could be much more predictable than the elm and ash tree due to their

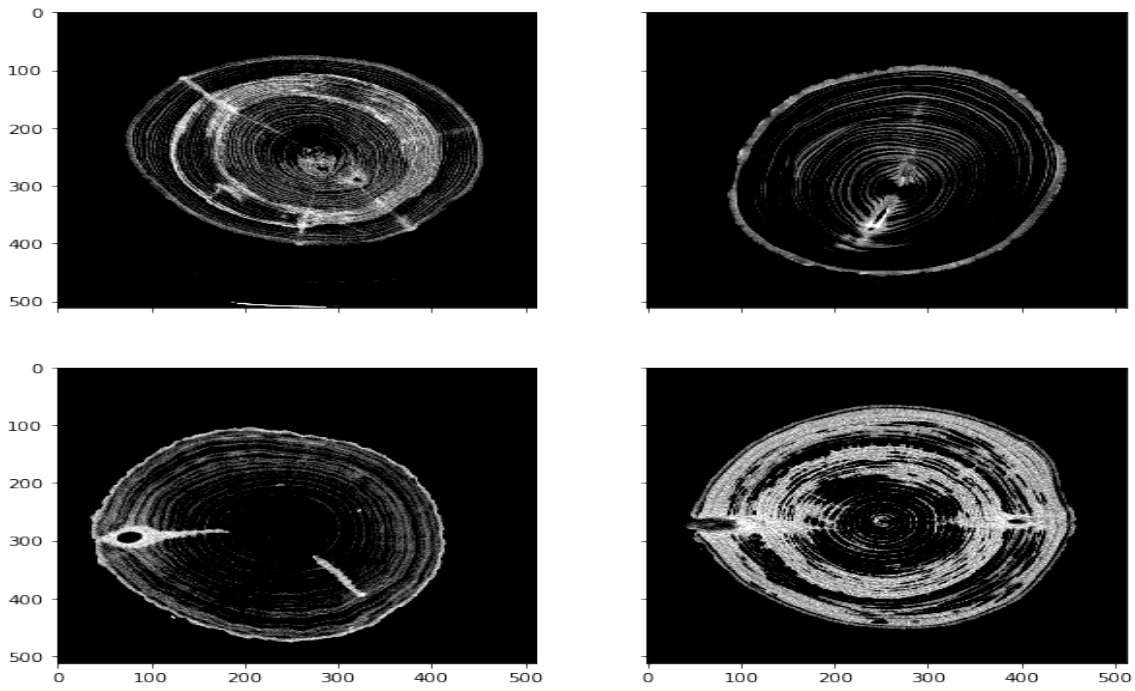

Figure 4.7: (Top left) elm tree (Top right) ash tree (Bottom left) aspen tree (Bottom right) fir tree

large side, in reality, the internal defects of the elm tree are much more correlated to the external surface. In fact, the ash, fir and aspen trees contains defects that don't emerge to the surface. We conclude that only the defects that have an impact on the surface have to be considered and trucked along the height of the tree. The other internal defects inside the cross section of the log must be generated based on the shape of their antecedents.

CHAPTER 5
CONCLUSION

Different neural network architectures have been compared to predict the inner density of synthetically made logs from their outer surface shape. Unlike 3D and CLSTM based models that are too complicated in terms of parameter number, 2D models are very simple encoder-decoder architectures with commonly used layers (Conv2D, Max-pooling2D, etc.). Thus, they are more robust to overfitting. They achieve better performance in terms of RMSE evaluation and visual quality assessment of the iso-surfaces than 3D-based models and CLSTM-based models. However, integrating the correlation assessment in density layer prediction (2D-models) with recurrent neural network approaches could give a better result. Generating a more look-like CT-scanned dataset of logs and their corresponding external surface provide us with valuable information about the necessity to identify and extract knots from the real CT-scanned cross-sections of the log. Hence, we are able to predict the distribution of key defects inside the round-wood of the tree from its external surface. However, the lack of real CT-scanned trees dataset affects the internal density prediction process. Further work should be the focus on the identification and extraction of the predictable defects inside the logs, but also on the exploration of 2D and 3D based model since they didn't achieve satisfactory prediction of the real CT-scanned internal density of the tree.

Appendices

APPENDIX A

EXPERIMENTAL EQUIPMENT

All experiments were carried out on an IBM Power Systems AC922 with 256 GB of RAM and 4 NVIDIA V100 16 GB GPGPU (using only a single GPGPU). Since on this platform, the GPGPU and the processors share a coherent memory space. We cannot guarantee that the previous settings are reproducible on Intel/AMD based machines as they may require smaller batches.

APPENDIX B

2D, 3D AND CONVLSTMS BASED MODEL PREDICTION ASSESSMENT

B.1 2D,3D and ConvLSTM model training-validation losses

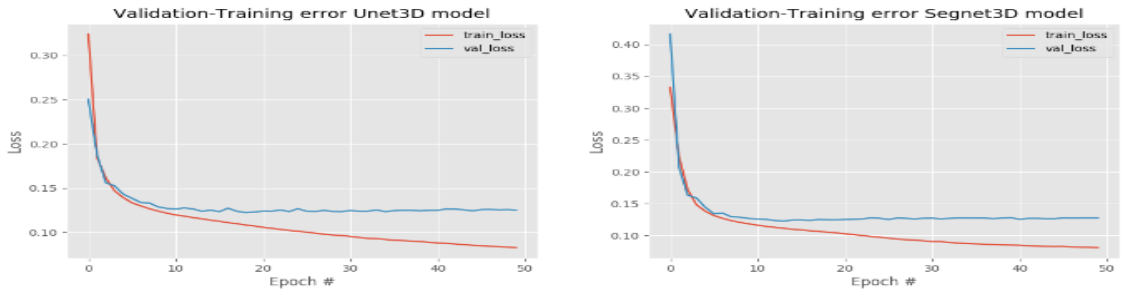

Figure B.1: 3D Models training validation losses

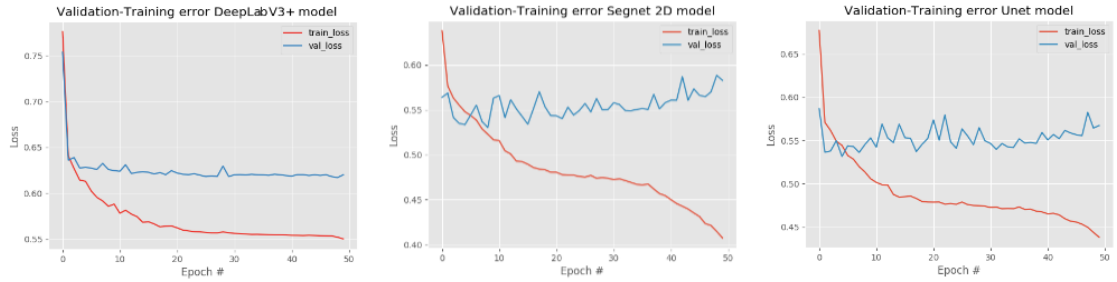

Figure B.2: 2D Models training validation losses

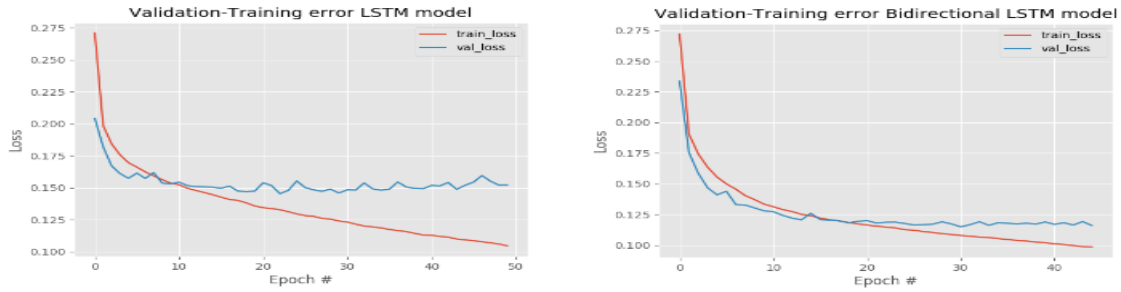

Figure B.3: CLSTM based Models training validation losses

B.2 2D,3D and ConvLSTM based model prediction assessment (RMSE)

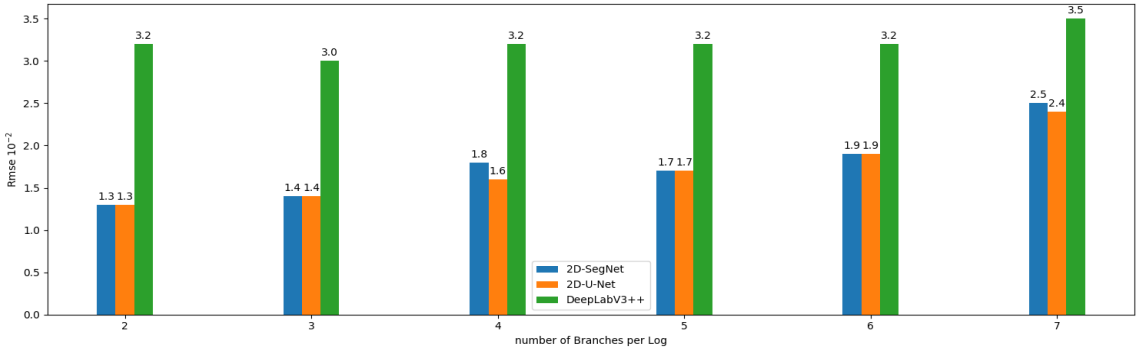

Figure B.4: RMSE between ground truth and predicted logs with 2D-models

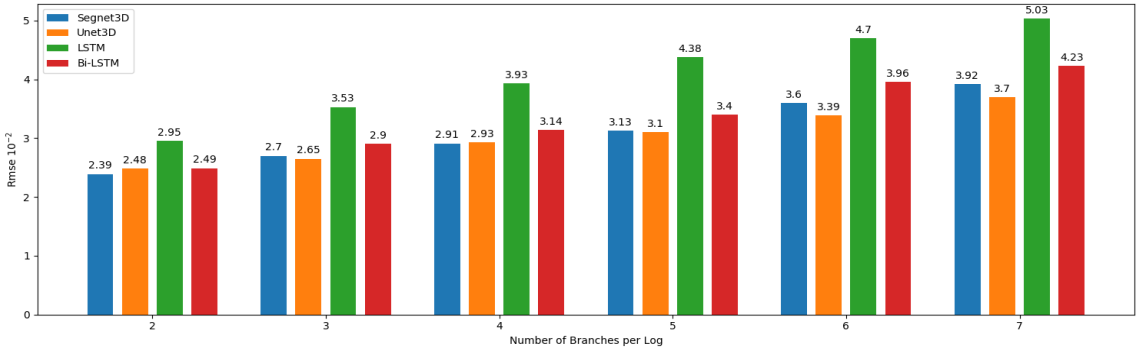

Figure B.5: RMSE between ground truth and predicted logs with 3D and ConvLSTM based models

B.3 external shape and internal structure of Real CT-Scanned trees

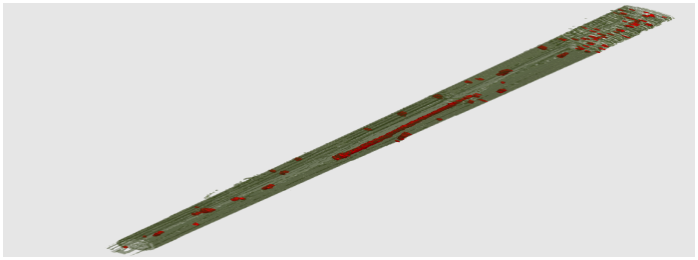

Figure B.6: 3D structure of an elm tree

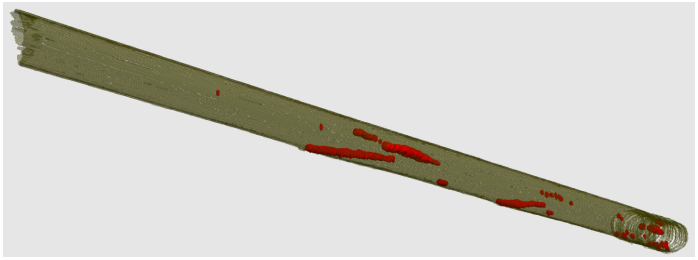

Figure B.7: 3D structure of an aspen tree

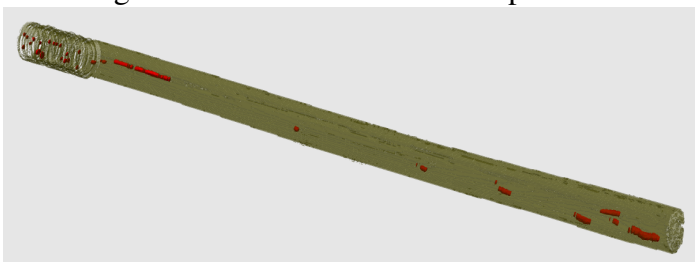

Figure B.8: 3D structure of an ash tree

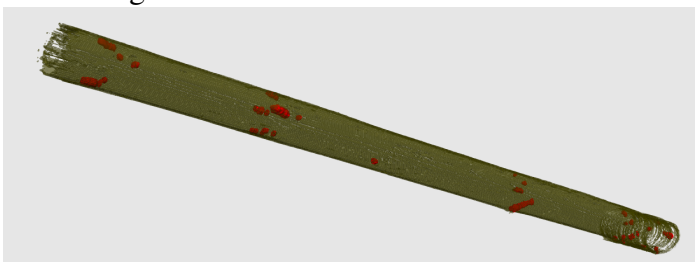

Figure B.9: 3D structure of an fir tree

REFERENCES

[1] M. Fredriksson, "Optimizing sawing of boards for furniture production using ct log scanning," 2015. eprint: 1409.1556.

[2] S. M. Stängle, F. Brüchert, A. Heikkila, T. Usenius, A. Usenius, and et al.., "Potentially increased sawmill yield from hardwoods using x-ray computed tomography for knot detection. annals of forest science," *Springer Verlag/EDP Sciences, 2015, 72 (1)*, pp. 57–65, 2015.

[3] C. A. Correa, M. R. Maldonado, D. M. Lozano, and C. A. Carrasco., "3d optimization of cutting patterns for logs of pinus radiata d.don with cylindrical defective core .," *MOSIM,10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nancy, France.*, 2014.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015. arXiv: 1511.00561 [cs.CV].

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. arXiv: 1505.04597 [cs.CV].

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018. arXiv: 1802.02611 [cs.CV].

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2014. arXiv: 1411.4038 [cs.CV].

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2016. arXiv: 1612.01105 [cs.CV].

[9] Özgün Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," 2016. arXiv: 1606.06650 [cs.CV].

[10] H. Chen, Q. Dou, L. Yu, and P.-A. Heng, "Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation," 2016. arXiv: 1608.05895 [cs.CV].

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. arXiv: 1512.03385 [cs.CV].

[12] S. shahabeddin Nabavi, M. Rochan, Yang, and Wang, "Future semantic segmentation with convolutional lstm," 2018. arXiv: 1807.07946 [cs.CV].

[13] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015. arXiv: 1506.04214 [cs.CV].

[14] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016. arXiv: 1603.07285 [stat.ML].

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2013. arXiv: 1311.2524 [cs.CV].

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015. arXiv: 1506.01497 [cs.CV].

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, 21–37, 2016.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. arXiv: 1409.1556 [cs.CV].

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015. arXiv: 1502.01852 [cs.CV].

[20] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. arXiv: 1405.0312 [cs.CV].